includes circuitry for bus interface units, memory interface units, buffers, and bus protocol logic. See the Abstract.

This is illustrated in Figure 1, where, for example, bus interface unit 140a interfaces data processing core 108 with CPU bus 100. See, for example, Figure 1 and column 4, line 65 through column 5, line 1.

Criteria for a Rejection under 35 U.S.C. § 102(b)

The criteria for a rejection under 35 U.S.C. § 102(b) has been clearly defined by the courts and confirmed by the U.S. Patent and Trademark Office. "A claim is anticipated only if each and every element as set Forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Independent Claim 1

Preamble of Claim 1

Claim 1 sets out an interface block that provides an interface between an internal bus and a socket of a logic block. The interface block, the internal bus and the logic block all are located within a single integrated circuit.

Examiner asserts that this is taught by Malladi. Particularly, Examiner argues that in Malladi, examples of an interface block are data processing shells 141a, 141b, 141c, 141b', 301,327 and 337.

Examiner asserts that in Malladi, an example of an internal bus is the CPU bus (i.e., CPU bus 100, 220 and 316).

Examiner additionally argues that in Malladi, an example of a socket of a logic block is memory shell 122, memory controller 322 or display controller unit 352 in Figures 1 though 3. This correlation made by Examiner is not correct. That is, Examiner has not correctly given examples of a socket of a logic block.

Specifically, memory shell 122, memory controller 322 and display controller unit 352 are all logic blocks that interact with a memory bus (i.e., memory bus 102 or 314). Data processing shells 141a, 141b, 141c, 141b', 301,327 and 337 are also logic blocks that interact with a memory bus (i.e., memory bus 102, 222 or 314). Thus, data processing shells 141a, 141b, 141c, 141b', 301, 327 and 337 do not provide an interface between the CPU bus (i.e., CPU bus 100, 220 and 316) and memory shell 122, memory controller 322 and display controller unit 352. Rather all these entities separately interface to memory bus 102.

Examiner's misreading of Malladi, allows for significant confusion in making correlations between the subject matter set out in the claims and the information disclosed by Malladi.

When correct correlations between Malladi and claim 1 of the present case are made, the analysis is much simpler and the differences are clearly seen.

For example, in Figure 1 of Malladi, data processing shell 141a provides two separate examples of interface blocks that provide an interface

between an internal bus and a logic block. Specifically data processing core 108 is the logic block. A bus interface unit 140a interfaces data processing core 108 with CPU bus 100. A memory interface unit 150a interfaces data processing core with memory bus 102.

Thus, if CPU bus 100 of Malladi is considered to be the internal bus set out in claim 1, then the interface block is bus interface 140a and the logic block is data processing core 108.

Examiner however, has asserted that when CPU bus 100 of Malladi is considered to be the internal bus set out in claim 1, then the interface block is processing shell 141a and memory shell 122 is the logic block. However, this is untenable. For example, processing shell 141a does not even interface with memory shell 122 but with memory bus 102. Rather, processing shell 141a and memory shell 122 are both separately connected to memory bus 102.

Elements of Claim 1

The elements of claim 1 are four separate modules: a synchronization module, a translation module, a queue module and a driver module.

Malladi discusses logic cells within a data processing shell; however, there is no one-to-one correspondence between the logic cells set out by Malladi and the four separate modules set out by claim 1 of the present case.

For example, claim 1 sets out a synchronization module that performs any needed synchronization between a clock domain of the internal bus and a clock domain of the socket of the logic block. Examiner has asserted this functionality is found in memory interface units 150a, 150b, 150c, 250b, 312,

326, or 336. However , memory interface units 150a, 150b, 150c, 250b, 312, or 326 do not just perform synchronization, but these memory interface units provide complete bus interface between a data processing core (108, 110, 112, etc.) and the memory bus (102, 222, 314). Thus any synchronization, translation, queuing or low level and electrical drive specifications functionality that needs to be performed between, for example, memory bus 102 and data processing core 108, is handled by memory interface unit 150a. Memory interface units 150a, 150b, 150c, 250b, 312, and 326 do not function merely as synchronization modules but operate as complete memory bus interfaces for their respective logic processing cores.

Likewise claim 1 sets out a translation module that, for data transferred between the internal bus and the socket of the logic block, provides translation of block encoding of the data. Examiner cites Malladi at column 5, lines 2 through 7 as disclosing this functionality. In this section, Malladi indicates that additional interface logic cells may represent staging buffers, local memory blocks, communications protocols etc. However, Malladi does not specifically disclose or suggest in this section (or any other section) a translation module that, for data transferred between an internal bus and a socket of a logic block, provides translation of block encoding of the data.

Likewise, claim 1 sets out a driver module that handles low level and electrical drive specifications of the internal bus. Examiner has asserted this functionality is found in bus interface units 140a, 140b, 140c, 240b, 310, 336, or 340. However , bus interface units 140a, 140b, 140c, 240b, 310, 336, and 340 do not just handle low level and electrical drive specifications, but these memory

interface units provide complete bus interface between a data processing core (108, 110, 112, etc.) and the CPU bus (100, 220, 316). Thus any synchronization, translation, queuing or low level and electrical drive specifications functionality that needs to be performed between, for example, CPU bus 100 and data processing core 108, is handled by bus interface unit 140a. Bus interface units 140a, 140b, 140c, 240b, 310, 336, and 340 do not function merely as driver modules but operate as complete memory bus interfaces for their respective logic processing cores.

Independent Claim 4

Preamble of Claim 4

Independent claim 4 sets out a method for providing an interface between an internal bus of an integrated circuit and a socket of a logic block within the integrated circuit.

As discussed above, Examiner argues that in Malladi, an example of a socket of a logic block is memory shell 122, memory controller 322 or display controller unit 352 in Figures 1 though 3. This correlation made by Examiner is not correct.

Specifically, memory shell 122, memory controller 322 and display controller unit 352 are all logic blocks that interact with a memory bus (i.e., memory bus 102 or 314). Data processing shells 141a, 141b, 141c, 141b', 301,327 and 337 are also logic blocks that interact with a memory bus (i.e., memory bus 102, 222 or 314). Thus, data processing shells 141a, 141b, 141c, 141b', 301,327 and 337 do not provide an interface between the CPU bus (i.e.,

CPU bus 100, 220 and 316) and memory shell 122, memory controller 322 and display controller unit 352. Rather all these entities separately interface to memory bus 102.

When correct correlations between Malladi and claim 4 of the present case are made, the differences between Malladi and the subject matter set out in claim 4 is evident.

For example, in Figure 1 of Malladi, data processing shell 141a provides two separate examples of interface blocks that provide an interface between an internal bus and a logic block. Specifically data processing core 108 is the logic block. A bus interface unit 140a interfaces data processing core 108 with CPU bus 100. A memory interface unit 150a interfaces data processing core with memory bus 102.

Thus, if CPU bus 100 of Malladi is considered to be the internal bus set out in claim 1, then the interface block is bus interface 140a and the logic block is data processing core 108.

Examiner however, has asserted that when CPU bus 100 of Malladi is considered to be the internal bus set out in claim 1, and memory shell 122 is the logic block. As discussed above, this is untenable.

Method Steps of Claim 4

In each of the steps of the method set out in claim 4, specific modules perform specific functions. In step (a), a synchronization module performs any needed synchronization between a clock domain of the internal bus and a clock domain of the socket of the logic block.

Examiner has asserted a synchronization module is disclosed by memory interface units 150a, 150b, 150c, 250b, 312, 326, or 336. However , memory interface units 150a, 150b, 150c, 250b, 312, and 326 do not just perform synchronization, but these memory interface units provide complete bus interface between a data processing core (108, 110, 112, etc.) and the memory bus (102, 222, 314). Thus any synchronization, translation, queuing, or low level and electrical drive specifications functionality that needs to be performed between, for example, memory bus 102 and data processing core 108, is handled by memory interface unit 150a. Memory interface units 150a, 150b, 150c, 250b, 312, and 326 do not function merely as synchronization modules but operate as complete memory bus interfaces for their respective logic processing cores.

In step (b), a translation module is used to provide any required translation of block encoding of data transferred between the internal bus and the socket of the logic block. Examiner cites Malladi at column 5, lines 2 through 7 as disclosing this functionality. In this section, Malladi indicates that additional interface logic cells may represent staging buffers, local memory blocks, communications protocols etc. However, Malladi does not specifically disclose or suggest in this section (or any other section) a translation module that, for data transferred between an internal bus and a socket of a logic block, provides translation of block encoding of the data.

In step (d), a driver module is used to handle low level and electrical drive specifications of the internal bus. Examiner has asserted a driver module is disclosed by bus interface units 140a, 140b, 140c, 240b, 310, 336, or

340. However , bus interface units 140a, 140b, 140c, 240b, 310, 336, and 340 do not just perform handle low level and electrical drive specifications, but these memory interface units provide complete bus interface between a data processing core (108, 110, 112, etc.) and the CPU bus (100, 220, 316). Thus any synchronization, translation, queuing, or low level and electrical drive specifications functionality that needs to be performed between, for example, CPU bus 100 and data processing core 108, is handled by bus interface unit 140a. Bus interface units 140a, 140b, 140c, 240b, 310, 336, or 340 do not function merely as driver modules but operate as complete memory bus interfaces for their respective logic processing cores.

Independent Claim 7

Independent claim 7 sets out an interface block that provides an interface between an internal bus of the integrated circuit and a socket of a logic block. The interface block comprises a plurality of modules connected in series. Each module in the pluralities performs only a single function from a plurality of functions. Any needed synchronization between a clock domain of the internal bus and a clock domain of the socket of the logic block is a first function from the plurality of functions. Any required translation of block encoding of data is a second function from the plurality of functions. Any buffering of data flowing between the internal bus and the socket of the logic block is a third function from the plurality of functions. Any low level and electrical drive specifications of the internal bus is a fourth function from the plurality of functions. This is not disclosed or suggested by Malladi.

Examiner has argued that in Malladi, an example of a socket of a logic block is memory shell 122, memory controller 322 or display controller unit 352 in Figures 1 though. This correlation made by Examiner is not correct. That is Examiner has not correctly given examples of a socket of a logic block.

Specifically, memory shell 122, memory controller 322 and display controller unit 352 are all logic blocks that interact with a memory bus (i.e., memory bus 102 or 314). Data processing shells 141a, 141b, 141c, 141b', 301,327 and 337 are also logic blocks that interact with a memory bus (i.e., memory bus 102, 222 or 314). Thus, data processing shells 141a, 141b, 141c, 141b', 301,327 and 337 do not provide an interface between the CPU bus (i.e., CPU bus 100, 220 and 316) and memory shell 122, memory controller 322 and display controller unit 352. Rather all these entities separately interface to memory bus 102.

Claim 7 sets out a plurality of modules connected in series. Each module in the plurality of modules performs only a single function from a plurality of functions. Examiner has argued that logic blocks 302, 304, 306, 308, 310 or 312 are examples of the plurality of modules connected in series. However, logic blocks 302, 304, 306, 308, 310 or 312 are not even connected in series and thus clearly do not qualify.

Claim 7 also variously asserts that memory interface units 150a, 150b, 150c, 250b, 312, and 326, and bus interface units 140a, 140b, 140c, 240b, 310, 336, and 340 perform the functions of modules in the plurality of modules. However, as discussed above, each of these interface units performs a complete interface between a logic core and a bus. None of these are a

module of an interface block that perform only a single function from a plurality of functions.

Dependent Claims

The dependent claims each disclose subject matter not disclosed or suggested by Malladi and in addition are allowable based on the allowability of their respective independent claims.

Conclusion

For all the reasons discussed above, Applicant believes that the present Application is in condition for allowance and favorable action is respectfully requested.

Respectfully submitted,
QUE-WON RHEE

By _____
Douglas L. Weller
Reg. No. 30,506

March 14, 2003
Santa Clara, California
(408) 985-0642